

Using API / Subprogram USR3005N

With this subprogram you can perform the following task:

- Read and maintain all documentation objects defined in Predict (predefined and user-defined).

This section contains:

- Calling the Subprogram
 - Predict DTD
 - Input to and Response from API / Subprogram USR3005N
 - Formulating Requests
 - Global and Special Attributes
-

Calling the Subprogram

Enter the command SYSEXT at the Natural prompt. This command invokes the library SYSEXT, which contains various Natural user exits. The following is provided for each user exit:

- a user exit subprogram (in object form)
- a sample program (in source form) of how to invoke the subprogram
- a text member explaining the function of the user exit
- several example programs that explain in detail how to read and maintain documentation objects.

A screen similar to the one below will appear. Page to subprogram USR3005N if necessary.

| | | | | |
|-----------|--------------------------------|-----------------|---|-----------------|
| 13:13:43 | ***** NATURAL USER EXITS ***** | | | 2003-05-31 |
| User: HNO | - Menu - | | | Library: SYSEXT |
| Cmd | Source | User Exit | Comment | Prod |
| - | USR1034P | USR1034N | Display NTTF file table | NAT |
| - | USR1035P | USR1035N | Maintain a NATURAL member in the source area | NAT |
| - | USR1036P | USR1036N | Maintain the user profile of the Software AG Edit | NAT |
| - | USR1037P | USR1037N | Information about NATURAL ABEND data | |
| - | USR1051P | USR1051N | Interface to various PRD data | PRD |
| - | USR2033P | USR2033N | Information about PRD LX sets | PRD |
| - | USR3005P | USR3005N | Process documentation objects | PRD |

Enter a question mark in the Cmd column to display a list of possible actions. Enter D in this column to display a description of this subprogram.

Predict DTD

```
<!ELEMENT Predict (Request | Result)>
<!ATTLIST Predict Version CDATA #IMPLIED>
<!ELEMENT Request (Select | Update | Purge | Add | Lock | Unlock)>
<!ELEMENT Select (Search, Return)>
<!ELEMENT Update (Search, Set)>
<!ELEMENT Purge (Search)>
<!ELEMENT Add ((Object-Type | Link), Set)>
<!ELEMENT Lock (Search)>
<!ELEMENT Unlock (Search)>
<!ELEMENT Result (Row * , Message)>
<!ELEMENT Message EMPTY>
```

```

<!ATTLIST Message number CDATA #REQUIRED
    text CDATA #REQUIRED
    type CDATA #REQUIRED
    invalid-attribute CDATA #IMPLIED
    index-in-invalid-attribute CDATA #IMPLIED
    additional-msg-number CDATA #IMPLIED
    additional-msg-text CDATA #OPTIONAL>
<!ELEMENT Search ((Object-Type | Link | Multi-Link), Attribute*)>
<!ATTLIST Search extract CDATA #IMPLIED
    from-date CDATA #IMPLIED
    key CDATA #IMPLIED
    key2 CDATA #IMPLIED
    key3 CDATA #IMPLIED
    key4 CDATA #IMPLIED
    key5 CDATA #IMPLIED
    key-not CDATA #IMPLIED
    key-op (AND | OR) #IMPLIED
    owner CDATA #IMPLIED
    scan-value CDATA #IMPLIED
    scan-abSTRACTS (Y | N) #IMPLIED
    scan-desc (Y | N) #IMPLIED
    scan-rules (Y | N) #IMPLIED
    scan-ignore-case (Y | N) #IMPLIED
    scan-obj-id (Y | N) #IMPLIED
    scan-absolute (Y | N) #IMPLIED>
<!ELEMENT Return (Field*)>
<!ELEMENT Set (Row *)>
<!ELEMENT Object-Type EMPTY>
<!ATTLIST Object-Type value CDATA #REQUIRED
    default CDATA #IMPLIED>
<!ELEMENT Link EMPTY>
<!ATTLIST Link source-object-type CDATA #REQUIRED
    association CDATA #REQUIRED
    direction (ACTIVE | PASSIVE) #REQUIRED>
<!ELEMENT Multi-Link EMPTY>
<!ATTLIST Multi-Link source-object-type CDATA #REQUIRED
    association CDATA #REQUIRED
    direction (ACTIVE | PASSIVE) #REQUIRED
    default CDATA #IMPLIED>
<!ELEMENT Attribute EMPTY>
<!ATTLIST Attribute name NMOKEN #REQUIRED
    value CDATA #REQUIRED
    display-only CDATA #OPTIONAL>
<!ELEMENT Field EMPTY>
<!ATTLIST Field name NMOKEN #REQUIRED>
<!ELEMENT Row (Attribute | Structure)*>
<!ELEMENT Structure (Group* | Attribute *)>
<!ATTLIST Structure name NMOKEN #REQUIRED
    display-only CDATA #OPTIONAL >
<!ELEMENT Group (Attribute *)>

```

Input to and Response from API / Subprogram USR3005N

Input to and response from the API uses XML documents. These XML documents must comply with the rules of the Predict DTD (document type definition) above.

Input to the API uses documents of the type Request while a response from the API uses documents of the type Result as defined in the first element of the Predict DTD. All data passed is represented as attribute(s) of an element. This method is used to reflect the dynamic extension of the Predict metastructure.

To analyze the result, you might use the copy code PARSER_X which is delivered in library SYSEXXT. The appropriate local data area PARSER-X also exists in library SYSEXXT.

Example

Search all Adabas files having the owner XYZ and a name starting with EMP and return some attributes.

```
<Predict>
  <Request>
    <Select>
      <Search>
        <Object-type value="FILE-A" Owner="XYZ" />
        <Attribute name="ID" value="EMP*" />
      </Search>
      <Return>
        <Field name="ID" />
        <Field name="CREATED-ON" />
        <Field name="CREATED-BY" />
        <Field name="CHANGED-ON" />
        <Field name="CHANGED-BY" />
        <Field name="ABSTRACT" />
      </Return>
    </Select>
  </Request>
</Predict>
```

The select request is used to read data from Predict. The `<Search>` element comprises a specification of the data to be retrieved (in the above example `<Object-type value="FILE-A" />`) and, if needed, a set of common search attributes (`Owner="XYZ"`, keyword, etc.) and optional additional search conditions formulated via the `<Attribute>` element (in the example `<Attribute name="ID" value="EMP*" />`).

All attributes that are to be returned in the result must be contained in the `<Return>` element. References to attributes not belonging to the addressed object type or association will not result in an error message but are simply skipped.

The result may look like this:

```
<Predict
  <Result>
    <Row>
      <Attribute Name="ID" Value="EMPLOYEES-FILE" />
      <Attribute Name="CREATED-ON" Value="199910130916489" />
      <Attribute Name="CREATED-BY" Value="PRD411" />
      <Attribute Name="CHANGED-ON" Value="199911161626520" />
      <Attribute Name="CHANGED-BY" Value="NATQA" />
      <Structure Name="Abstract">
        <Attribute Name="Abstract-line" Value="CONTAINS UNIQUE DESCRIPTIONS" />
        <Attribute Name="Abstract-line" Value="OF EMPLOYEES OF AN INTERN." />
        <Attribute Name="Abstract-line" Value="ENTERPRISE (COMPAR. TO SAG)" />
      </Structure>
    </Row>
    <Row>
      <Attribute Name="ID" Value="EMPLOYEES-ABOVE" />
      <Attribute Name="CREATED-ON" Value="199807071515436" />
      <Attribute Name="CREATED-BY" Value="NATQA" />
      <Attribute Name="CHANGED-ON" Value="199807141258276" />
      <Attribute Name="CHANGED-BY" Value="RRI" />
      <Structure Name="Abstract">
        <Attribute Name="Abstract-line" Value="This file was incorporated" />
        <Attribute Name="Abstract-line" Value="from FDT on 98-07-07" />
      </Structure>
    </Row>
  </Result>
</Predict>
```

```

</Row>
<Message number="2517" text="DIC2517 Function TERMINATED SUCCESSFULLY." type="Success"/>
</Result>
</Predict>

```

The `type` attribute of the `<Message>` tag indicates whether the request could be successfully processed. Every object in the result document is delimited by a `<Row>` element.

Formulating Requests

Select Requests

You specify an object type with the `<Object-type>` element of the `<Search>` element. Additional search conditions can be given with the `<Attribute>` elements of the `<Search>` element, by specifying the attribute ID and the corresponding search value. For alphanumeric attributes, asterisk notation is allowed to search for all objects that have an attribute value which starts with the given search value.

Attribute values are transferred as strings. Both the quote character and the double-quote character can be used as string delimiters. Quotes within values must be coded as '`'`'.

All replacements:

| | |
|---|-------------------------|
| & | <code>&amp;</code> |
| ' | <code>&apos;</code> |
| " | <code>&quot;</code> |
| < | <code>&lt;</code> |
| > | <code>&gt;</code> |

With the `return` element, the IDs of implicitly or explicitly defined attributes are specified, for which the values are to be returned. Only attributes belonging to the given object type may be specified, the others are ignored.

Note:

When reading or modifying periodic groups, multiple value fields or simple groups additional rules should be considered. Special restrictions apply for fields with format logical or numeric. See Object Type #ATTRIBUTE for further information.

Modification Requests

To allow updating or deleting of an object, it must first be locked. An update request does not unlock the object. This has to be requested explicitly. A successful delete request will also remove the lock for the object, whereas an unsuccessful delete request will keep the lock.

Links and multiple links are considered to be attributes of the parent object. This implies that the parent object of a link or multiple link has to be locked before the link/multiple link can be modified.

Update and delete requests may only refer to one object at a time. That means that the result of the search operation consists only of one object.

Files have an attribute named `Element List`. With this attribute, the base attributes of all fields belonging to a file are transferred. There is no Add function for fields of a file. To add new fields to a file, the complete element list has to be updated.

There is no explicit rename request. Renaming must be done by updating the attribute ID. The object must be unlocked with the new ID.

To add exactly one link to a link list, use the <Add> element in your search request.

To delete exactly one link from a link list, use the <Purge> element in your search request.

If you want to add or modify the association attributes of only a single link in a link list by using the <Update> element, then your search request must include a fully qualified attribute of type Target-Object.

If you want to modify the complete link list of the associated object (including all association attributes) by using the <Update> element, then a fully qualified attribute of type Target-Object is not necessary in your search request.

Link a program to a system.

```
<Predict>
  <Request>
    <Add>
      <Link source-object-type="SYSTEM" association="PROGRAM" direction="ACTIVE" />
    <Set>
      <Row>
        <Attribute name="SOURCE-OBJECT" value="SYSTEM1" />
        <Attribute name="TARGET-OBJECT" value="PROGRAM1" />
      </Row>
    </Set>
  </Add>
</Request>
</Predict>
```

Example

Update some Adabas attributes of a file.

```
<Predict>
  <Request>
    <Update>
      <Search>
        <Multi-Link source-object-type="DATABASE-A" association="FILE" direction="ACTIVE" />
        <Attribute name="SOURCE-OBJECT" value="DB180" />
        <Attribute name="TARGET-OBJECT" value="CHD-A-FORMATE" />
      </Search>
    <Set>
      <Row>
        <Attribute Name="MAX-ISN" Value="800" />
        <Attribute Name="ASSO-DEVICE-TYPE" Value="3390" />
        <Attribute Name="DATA-DEVICE-TYPE" Value="3390" />
        <Attribute Name="ASSOPFAC" Value="91" />
        <Attribute Name="CIPHERED" Value="Y" />
      </Row>
    </Set>
  </Update>
</Request>
</Predict>
```

Fields can only be added by modifying the element list of a file. Modifications to the field attributes type, level, format and length can also be made in the element list only.

The element list has an attribute named EL-UNIQUE-ID. When updating the element list, the value of this attribute indicates whether a field has been added, modified or renamed.

General

With this version it is not possible to change the sub-type of an object (e.g. change a conceptual file into an Adabas file). It is not possible to maintain the Predict metadata.

Predict security checks are done as usual.

The Predict user exits U-ACMR, U-DESC, U-MNT, U-MNT1, U-OW, U-PUR and U-SEC are invoked as defined.

Global and Special Attributes

The following global attributes are available for objects of all object types defined on the FDIC. The notation (S) after an attribute name means that this attribute can be used in a search condition:

- ID (S)
- CREATED-BY
- CREATED-ON
- CHANGED-BY
- CHANGED-ON
- ABSTRACT (representing a structure with the attribute ABSTRACT-LINE)
- KEYWORDS (representing a structure with the attribute KEYWORD)
- OWNERS (representing a structure with the attribute OWNER)
- DESCRIPTION (representing a structure with the attribute DESCRIPTION-LINE)
- OBJECT-STATUS (possible values are OBJECT, DUMMY and PLACEHOLDER)
- #OBJECT-TYPE. The value of this attribute represents the specialization of the object type (for example, "FILE-A"). If there is no specialization, the value of the base type is represented.
- #BASE-TYPE. The value of this attribute represents the name of the base type (for example "FILE").
- #OBJECT-TITLE. The value of this attribute represents the title of the specialization of the object type (for example, "Adabas file").
- #BASE-TITLE. The value of this attribute represents the title of the base type (for example, "File").

Object Type #ATTRIBUTE

To provide attribute definitions specific to an object type or to an association, the special object type #ATTRIBUTE exists. The notation (S) after an attribute name means that this attribute can be used in a search condition.

Objects of the type #ATTRIBUTE can have the following attributes:

- ID (S)
- TITLE
- NAMESPACE (S)
- NAMESPACETYPE (S) (either #OT-DEFINITION or #ASSOCIATION)
- NAMESPACE-START-OBJECTTYPE (S) (only for Namespacetype="#ASSOCIATION")
- NAMESPACE-DIRECTION (S) (only for Namespacetype="#ASSOCIATION")
- BELONGS-TO-SCREEN (S)
- IS-SCREEN (S) (Valid values: Y or N)
- LEVEL (Used to group attributes)
- FORMAT
 - A - Alphanumeric
 - N - Numeric
 - D - Date
 - T - Time
 - L - Logic

- X - Literal
- E - Text
- F - Frame (Group frame, if level = 1, it denotes a tab/screen)
- S - Structure
- M - Field length with unit specification
- LENGTH
- MULTIPLE (N5)
- SEARCHABLE (S) (Indicates that this attribute can be used in search conditions.)

Example

Get all attributes of the object type application library.

```
<Predict>
  <Request>
    <Select>
      <Search>
        <Object-type value="#ATTRIBUTE" />
        <Attribute name="NAMESPACETYPE" value="#OT-DEFINITION" />
        <Attribute name="NAMESPACE" value="SYSTEM-A" />
      </Search>
      <Return>
        <Field name="TITLE" />
        <Field name="ID" />
        <Field name="FORMAT" />
        <Field name="LENGTH" />
      </Return>
    </Select>
  </Request>
</Predict>
```

The result may look like this:

```
<Predict>
  <Result>
    <Row>
      <Attribute Name="TITLE" Value="Implementation Pointer" />
      <Attribute Name="ID" Value="IMPLEMENTATION-POINTER" />
      <Attribute Name="FORMAT" Value="F" />
      <Attribute Name="LENGTH" Value="0" />
    </Row>
    <Row>
      <Attribute Name="TITLE" Value="Library" />
      <Attribute Name="ID" Value="LIBRARY" />
      <Attribute Name="FORMAT" Value="A" />
      <Attribute Name="LENGTH" Value="8" />
    </Row>
    <Row>
      <Attribute Name="TITLE" Value="User system Fn" />
      <Attribute Name="ID" Value="FUSER-FNR" />
      <Attribute Name="FORMAT" Value="N" />
      <Attribute Name="LENGTH" Value="5" />
    </Row>
    <Row>
      <Attribute Name="TITLE" Value="User system DBnr" />
      <Attribute Name="ID" Value="FUSER-DBID" />
      <Attribute Name="FORMAT" Value="N" />
      <Attribute Name="LENGTH" Value="5" />
    </Row>
    <Row>
      <Attribute Name="TITLE" Value="Type" />
```

```

<Attribute Name="ID" Value="SUBTYPE" />
<Attribute Name="FORMAT" Value="A" />
<Attribute Name="LENGTH" Value="1" />
</Row>
<Message number="2517" text="Function TERMINATED SUCCESSFULLY." type="Success" />
</Result>
</Predict>

```

If an attribute defined with the format S (Structure) or F (Frame) includes more than one attribute and `Multiple` has a value > 0, then it is transferred with the `<Structure>` element and the included attributes have to be grouped with the `<Group>` element. The structure represents a periodic group. The sequence of the `<Group>` elements represents the occurrences.

If an attribute defined with the format S (Structure) or F (Frame) includes exactly one attribute and `Multiple` has a value > 0, then it is transferred with the `<Structure>` element and no `<Group>` elements are transferred. The structure or the included attribute represents a multiple value field.

If an attribute defined with the format S (Structure) or F (Frame) includes more than one attribute and `Multiple` has the value 0, it is transferred with the `<Structure>` element and the included attributes are not grouped. The structure represents a simple group.

Values for attributes with the format L (Logic) are transferred as string with content 'Y' or 'N'.

Numeric values with decimal precision (Format N and M) are transferred without a decimal point. For example, the value 75.3 for an attribute defined with Length 5 and Precision 1 is transferred as '753'. For an attribute defined with Length 5 and Precision 2, the value '7530' has to be specified.

Object Type #OT-DEFINITION

To provide all object types defined on the FDIC, the special object type #OT-DEFINITION exists. The notation (S) after an attribute name means that this attribute can be used in a search condition.

Objects of type #OT-DEFINITION can have the following attributes:

- ID (S)
- TITLE
- TYPE (S) (Object type or specialization type. If the value "ALLOBJECTTYPE" is used, the result list includes the object type Field.)
- CREATED-BY
- CREATED-ON
- CHANGED-BY
- CHANGED-ON
- INSTANCE-EXIST (Indicates if objects of the specified type exist.)

Objects of type #OT-DEFINITION have an indicator named "Type" that shows whether this object is a definition of a base object type (e.g. Database, File, Program, etc.) or of a specialization type (e.g. Adabas database, DB2 database, Adabas file, DB2 table, Subprogram, Subroutine, etc.).

Base object types have only those attributes common to all objects of this type. Specialization types additionally have those attributes that are specific to objects of the specified sub-type.

Example

Search for all object types defined on the FDIC and, for each, return the title and the ID.

```
<Predict>
  <Request>
    <Select>
      <Search>
        <Object-type value="#OT-DEFINITION" />
        <Attribute name="TYPE" value="OBJECTTYPE" />
      </Search>
      <Return>
        <Field name="TITLE" />
        <Field name="ID" />
      </Return>
    </Select>
  </Request>
</Predict>
```

The result may look like this:

```
<Predict>
  <Result>
    <Row>
      <Attribute Name="TITLE" Value="Keyword" />
      <Attribute Name="ID" Value="KEYWORD" />
    </Row>
    <Row>
      <Attribute Name="TITLE" Value="Network" />
      <Attribute Name="ID" Value="NETWORK" />
    </Row>
    .
    .
    .
    <Row>
      <Attribute Name="TITLE" Value="File" />
      <Attribute Name="ID" Value="FILE" />
    </Row>
  <Message number="2517" text="Function terminated successfully." type="Success" />
</Result>
</Predict>
```

Object Type #ASSOCIATION

To provide all associations defined on the FDIC, the special object type #ASSOCIATION exists.

Objects of type #ASSOCIATION can have the following attributes:

- ID (S)
- Title
- SOURCE-OBJECT-TYPE (S)
- SOURCE-TYPE-TITLE
- TARGET-OBJECT-TYPE (S)
- TARGET-TYPE-TITLE
- DIRECTION (S) (Indicates the direction of the association. Valid values: ACTIVE or PASSIVE.)
- MANDATORY (S)
- CREATED-BY
- CREATED-ON
- CHANGED-BY
- CHANGED-ON
- ABSTRACT
- ADDITIONAL-ATTRIBUTES (Valid values: Y or N.)
- MAINTAINABLE (by User)

- MULTI-LINKS-POSSIBLE

Example

Search for all active associations of Adabas files:

```
<Predict>
  <Request>
    <Select>
      <Search>
        <Object-type value="#ASSOCIATION"/>
        <Attribute name="SOURCE-OBJECT-TYPE" value="FILE-A"/>
        <Attribute name="DIRECTION" value="ACTIVE"/>
      </Search>
      <Return>
        <Field name="TITLE"/>
        <Field name="ID"/>
        <Field name="TARGET-OBJECT-TYPE"/>
        <Field name="TARGET-TYPE-TITLE"/>
      </Return>
    </Select>
  </Request>
</Predict>
```

The result may look like this:

```
<Predict>
  <Result>
    <Row>
      <Attribute Name="TITLE" Value="Has Fields"/>
      <Attribute Name="ID" Value="ELEMENT"/>
      <Attribute Name="TARGET-OBJECT-TYPE" Value="ELEMENT"/>
      <Attribute Name="TARGET-TYPE-TITLE" Value="Field"/>
    </Row>
    <Row>
      <Attribute Name="TITLE" Value="Linked to VE"/>
      <Attribute Name="ID" Value="LINKED_TO_VE"/>
      <Attribute Name="TARGET-OBJECT-TYPE" Value="VERIFICATION"/>
      <Attribute Name="TARGET-TYPE-TITLE" Value="Verification"/>
    </Row>
    <Row>
      <Attribute Name="TITLE" Value="Has Direct Views"/>
      <Attribute Name="ID" Value="HAS_DIRECT_VIEWS"/>
      <Attribute Name="TARGET-OBJECT-TYPE" Value="FILE"/>
      <Attribute Name="TARGET-TYPE-TITLE" Value="File"/>
    </Row>
    <Row>
      <Attribute Name="TITLE" Value="Has Sequence EL"/>
      <Attribute Name="ID" Value="HAS_SEQUENCE_EL"/>
      <Attribute Name="TARGET-OBJECT-TYPE" Value="ELEMENT"/>
      <Attribute Name="TARGET-TYPE-TITLE" Value="Field"/>
    </Row>
    <Message number="2517" text="Function terminated successfully." type="Success"/>
  </Result>
</Predict>
```

LINK and MULTI-LINK

To allow searching for links between objects, the <Link> and <Multi-Link> elements exist.

Links can have the following attributes:

- SOURCE-OBJECT (S)
- TARGET-OBJECT (S)
- SOURCE-OBJECT-TYPE
- SOURCE-TYPE-TITLE
- TARGET-TYPE-TITLE
- TARGET-OBJECT-TYPE
- TARGET-OBJECT-STATUS
- MULTIPLE-LINK (Valid values: Y or N. Indicates whether additional attributes are provided via link or multi-link.)
- MULTIPLE-ALLOWED (Valid values: Y or N. Indicates whether a child object can be linked more than once to the same parent object via the same association. Currently used for FI-ADA only.)

In addition, links from fields have the following searchable attribute:

- SOURCE-OBJECT-NAMESPACE (S)

In addition, links to fields have the following searchable attribute:

- TARGET-OBJECT-NAMESPACE (S)

Furthermore, links can have additional attributes that have been defined for the corresponding association.

When searching for links, if the name of a specialization is used as input for SOURCE-OBJECT-TYPE, than the sub-type is ignored and the base object type is used.

For example, it is possible to search for links to files from the conceptual database A-C with the following search request:

```
<Search>
  <Link source-object-type value="DATABASE-A" association="FILE" Direction="ACTIVE" />
  <Attribute name="SOURCE-OBJECT" value="A-C" />
</Search>
```

Example

Search for all programs linked to the conceptual system SAG-PRD-OTHER-PGMS via the active association PROGRAM:

```
<Predict>
  <Request>
    <Select>
      <Search>
        <Link source-object-type="SYSTEM-C" association="PROGRAM" direction="ACTIVE" />
        <Attribute name="SOURCE-OBJECT" value="SAG-PRD-OTHER-PGMS" />
      </Search>
      <Return>
        <Field name="TARGET-OBJECT" />
        <Field name="TARGET-OBJECT-STATUS" />
        <Field name="TARGET-OBJECT-TYPE" />
        <Field name="TARGET-TYPE-TITLE" />
        <Field name="MULTIPLE-LINK" />
        <Field name="MULTIPLE-ALLOWED" />
      </Return>
    </Select>
  </Request>
</Predict>
```

The result may look like this:

```
<Predict>
  <Result>
    <Row>
      <Attribute Name="TARGET-OBJECT" Value="SAG-PRD-SUMPRDEX" />
      <Attribute Name="TARGET-OBJECT-STATUS" Value="OBJECT" />
      <Attribute Name="TARGET-OBJECT-TYPE" Value="PROGRAM-N" />
      <Attribute Name="TARGET-TYPE-TITLE" Value="Subprogram" />
      <Attribute Name="MULTIPLE-LINK" Value="N" />
      <Attribute Name="MULTIPLE-ALLOWED" Value="N" />
    </Row>
    <Row>
      <Attribute Name="TARGET-OBJECT" Value="SAG-ADABAS" />
      <Attribute Name="TARGET-OBJECT-STATUS" Value="OBJECT" />
      <Attribute Name="TARGET-OBJECT-TYPE" Value="PROGRAM-E" />
      <Attribute Name="TARGET-TYPE-TITLE" Value="External program" />
      <Attribute Name="MULTIPLE-LINK" Value="N" />
      <Attribute Name="MULTIPLE-ALLOWED" Value="N" />
    </Row>
    <Message number="2517" text="Function terminated successfully." type="Success" />
  </Result>
</Predict>
```

Multi-links can have the following attributes:

- SOURCE-OBJECT (S)
- TARGET-OBJECT (S)
- SOURCE-TYPE-TITLE
- TARGET-TYPE-TITLE
- CREATED-BY
- CREATED-ON
- CHANGED-BY
- CHANGED-ON

Note:

Multi-links must have one additional identifying attribute of the associated link. For example, the link *Has files* for the object type Database (FI-ADA) has the additional identifying attribute P-FNR.